

[View](#) [Sync](#) [Diff Revs](#) [Diff vs ClientFile](#) [Describe Changelist](#) [Help](#)

Revision	File Name	Changelist	Date	Description
----------	-----------	------------	------	-------------

6	//depot/m...	6283	2000/01/28	Switched ical away from factory, ripped some factory code out of calendar
5	//depot/m...	6077	2000/01/25	change UID to be the DbRecID instead of a separate field
4	//depot/m...	2273	2000/01/13	Update for receiving sync requests
3	//depot/m...	2120	2000/01/10	Update ical sync, get receive sync requests to work
2	//depot/m...	1973	2000/01/05	Files that have changed between 1.7 and 2.0
1	//depot/m...	1955	2000/01/05	Applications 1.7 material

Actions

## 1

1

```

45         public:
            static TIME tAlarmTrigger;

            CalendarDatabase( TASK &taskId, char *fileName, DatabaseBusyStatus* dbsPtr,byte version = 0 ) :
            Database( fileName, dbsPtr, version ), {(taskId)
            {
                check_database();
            }

            virtual bool backup_restore_completed();
            virtual void backup_restore_started();

            // for future expansion, when we have incompatible versions....
            virtual void incompatible_record_version( DbRecordHandle db_record_handle, byte version )
            {
                Database::incompatible_record_version ( db_record_handle, version );
            }

            virtual bool delete_record( DbRecordHandle db_record_handle );
        };

65     extern CalDLLAccess CalendarDatabase calendarDatabase;

70

const int NO_ALARM_FLAG = 0x80;

75     const DbFieldTag TYPE_FIELD_ID           = 1;
        const DbFieldTag TITLE_FIELD_ID       = 2;
        const DbFieldTag TEXT_FIELD_ID        = 3;
        const DbFieldTag LOCATION_FIELD_ID    = 4;
        const DbFieldTag TRIGGER_FIELD_ID     = 5;
        const DbFieldTag STARTS_FIELD_ID      = 6;
        const DbFieldTag ENDS_FIELD_ID        = 7;
        const DbFieldTag IS_DUE_FIELD_ID      = 8;
        const DbFieldTag STATUS_FIELD_ID      = 9;
        const DbFieldTag PRIORITY_FIELD_ID    = 10;
        const DbFieldTag ATTENDEE_FIELD_ID    = 11;
        const DbFieldTag RECURRENCE_FIELD_ID = 12;

        // type of event: appointment, reminder, note, journal, task
        // title of event
        // details about event(text body)
        // location of event
        // alarm trigger
        // start date/time of event
        // end date/time of event
        // date/time task due - Not used
        // status of task - Not Used
        // priority of task - Not Used
        // list of attendees to event
        // Recurring info.

```

# CalDb.h

```

const DbFieldTag EXCLUSIONS_FIELD_ID = 13;
//const DbFieldTag RECORD_UID_ID = 14;
const DbFieldTag CALENDAR_UID_ID = 15;
const DbFieldTag SEQUENCE_DEVICE_FIELD_ID = 16;
const DbFieldTag SEQUENCE_HOST_FIELD_ID
const DbFieldTag RELATED_TO_FIELD_ID
const DbFieldTag ATTENDEE_DECLINED_FIELD_ID
const DbFieldTag ATTENDEE_TENTATIVE_FIELD_ID = 20;
const DbFieldTag ATTENDEE_NEEDSACTION_FIELD_ID = 21;
const DbFieldTag ORGANIZER_FIELD_ID = 22;
const DbFieldTag RECURRENCE_UID_ID = 23;
const DbFieldTag SEQUENCE_SENT_FIELD_ID = 24;

const DbFieldTag ALLDAY_FIELD_ID = 255;

// Exclusion field list
// Unique ID (assigned by device (+) or host (-))
// Unique ID (assigned by host calendar program)
// Sequence id of last update from device
// Sequence id of last update from host
// the record uid of parent recurrence event
// list of declined attendees to event
// list of tentative attendees to event
// list of attendees who have not responded
// organizer field
// uniquely identifies a single instance of a recurring event
// the last sequence successfully sent to the host

const DbFieldTag ALLDAY_FIELD_ID = 255; // Reference Only for Puma Sync

// xxx_AND_CLASS repeat patterns
enum
{
    ANY_DAY = 7,
    ANY_WEEKDAY = 8,
    ANY_WEEKEND_DAY = 9,
};

// error flags:
const int INVALID_TYPE = -1;
const int INVALID_CHOICE = -1;

// calendar event types:
const char NOTE_RECORD = 'n';
//const char SCHEDULE_RECORD = 's';
//const char JOURNAL_RECORD = 'j';
const char EVENT_RECORD = 'a';
const char TASK_RECORD = 't';
//const char MULTIPLE_RECORDS = '*';
const char RECURRING_RECORD = '*';

// calendar event types:

#pragma pack(push, 1)
typedef struct _RepeatInfo
{

```